SYSTEM AND METHOD FOR DYNAMIC CLOCK GENERATION

INVENTORS

EDMUND CHEUNG
3530 Murdoch Dr.
Palo Alto, CA 94036

OTTO SPONRING
1013 Russell Ave.
Los Altos, CA 94024

BACKGROUND

FIELD OF THE INVENTION

This invention pertains in general to portable electronic devices and in particular to reducing power consumption of such devices by controlling the clock frequencies in the devices.

BACKGROUND OF THE INVENTION

The need for portable electronic devices, also referred to as "electronic appliances" is growing rapidly. Numerous small devices, such as the PALM PILOT organizer from 3COM, INC. and MICROSOFT WINDOWS CE-based palm computers have become ubiquitous in society. As the sizes of the devices decrease and the processing powers increase, these devices will become even more prevalent. For example, solid-state music players, smart phones, screen phones, digital cameras, and other Internet-ready electronic appliances will soon become mainstream.

By nature, a portable electronic device should be as small and light as possible. Accordingly, there is a desire to reduce the amount of circuitry within the device by combining functions previously performed by separate integrated circuits (ICs) into a single complementary metal oxide semiconductor (CMOS) application specific integrated circuit (ASIC). For example, a single ASIC for a portable electronic device may include a central processing unit (CPU), a digital signal processor (DSP), a peripheral controller, a memory controller, a video controller, a clock controller, and an interrupt controller.

However, the CMOS ASIC requires a significant amount of power to provide this functionality. Since the battery is often one of the heaviest and bulkiest components of a

portable electronic device, there is a strong desire to minimize power consumption by the ASIC and other components. Although a single ASIC typically uses less power than do separate ICs, there is a desire to reduce the power use even further.

For example, in a CMOS ASIC, the majority of power dissipation is due to the alternating current (AC) element that results from the charging and discharging of the capacitance in the chip. The power dissipation of a typical logic gate in a CMOS ASIC is approximated by the following equation:

Power Dissipation = $V_{dd}^2 * C * Freq$;

where V_{dd} is the voltage supply to a logic gate, C is the total intrinsic and extrinsic capacitance loading of the logic gate, and Freq is the toggle frequency of the logic gate. The total power dissipation of the CMOS ASIC is the sum of all power dissipated at each logic gate at the gate's respective toggle frequency and total capacitance loading.

Thus, the major power draws on the ASIC are the clocks generated by the clock controller and the digital logic that is controlled directly or indirectly by the clocks. A typical clock controller, for example, may generate separate clocks for driving the CPU, system bus, memory, and peripherals. The operating speed of the driven device is dependent on the frequency of its clock. Since there is a general desire to operate the device at its highest speed, the clocks are usually run at a high frequency and therefore dissipate power at a high rate.

Moreover, if a device is operated with a clock frequency higher than is necessary, the power utilized to operate at the higher-frequency clock is essentially wasted. For example, the system memory may run on a 100 MHz clock. However, an external peripheral accessing the memory, such as an infra-red (IR) transceiver, may support data

transfers at only a fraction of the speed of which the memory is capable. Assuming that there are no concurrent memory requests from other devices, the power needed to drive the system memory at a higher frequency than is needed to support the IR transceiver is essentially wasted.

Some ASICs contain special circuitry to reduce power consumption by disabling external memory devices using the "CKE" signal when there are no memory accesses. While the external memory devices save power by entering a low power state when disabled, the memory clock generated by the clock controller remains running at full speed. In addition, the "Clock I/O" pin to the external memory also toggles continuously, thereby consuming power on the clock driver I/O pad and the clock receiver I/O pads.

Accordingly, there is a need for a low-power CMOS ASIC for a portable electronic device. Preferably, the ASIC would minimize power wasted by free running clocks and mismatches between the bandwidth requirements of different devices. The ASIC should also achieve these goals without sacrificing system performance.

SUMMARY OF THE INVENTION

The above problems are resolved by an application specific integrated circuit (ASIC) for a portable electronic device that has a clock controller that dynamically and automatically varies the frequency of on-chip clocks in response to bandwidth requirements of the driven logic. The ASIC preferably includes a central processing unit (CPU), a digital signal processor, and a small amount of static random access memory (SRAM). In addition, the ASIC includes system and memory busses coupling together the CPU, a memory controller for accessing external memory devices, controllers for

supporting color and/or monochrome display devices, an interrupt controller, and a direct memory access (DMA) controller for providing other devices access to the memory controller without requiring CPU intervention. The ASIC also preferably includes a peripheral bus coupling other peripherals to the system bus. Devices on the peripheral bus include universal asynchronous receiver/transmitters (UARTs), MMC card interfaces, and synchronous serial interfaces. The peripheral bus is also directly connected to the DMA controller.

The ASIC also includes one or more oscillators used by phase locked loops (PLLs) to generate one or more master clocks. These master clocks are received by a system clock controller which derives various clocks of different frequencies from the master clocks. These derived clocks are used to drive the various controllers and peripherals described above. For example, the system clock controller preferably generates a memory clock for clocking the memory controller and the external memory devices, a bus clock for clocking the system bus, a CPU clock for clocking the CPU, and one or more peripheral clocks for clocking the various peripheral controllers and peripherals coupled to the ASIC.

The various devices in the ASIC that can be accessed by other devices in the ASIC, such as the system bus, the memory controller, and the SRAM, are referred to as "resources." The speed at which a resource is clocked affects the rate at which the resource can process data (i.e., the bandwidth of the resource). The devices in the ASIC that can access a resource are referred to as "controllers." For example, a UART controller is a controller because it can access the memory controller via the DMA controller. Typically, a controller will access a resource within a given bandwidth range.

Preferably, every controller has a request line coupled to the system clock controller to indicate when the controller is accessing a resource. The controller may have a single request line to indicate that it is accessing any resource, or the controller may have a request line for each resource in the ASIC that the controller can access. In addition, the system clock controller has a programmable bandwidth register associated with each controller for holding a value representing the bandwidth utilized by the controller. The system clock controller also preferably includes an adder, a frequency table, and a multiplexer (MUX) for each clocked resource.

When a controller accesses a resource, the controller signals the system clock controller via the request line. The system clock controller, in turn, uses the adder to sum the values held in the bandwidth registers of all of the controllers that are currently accessing the resource. The resulting sum is then used an index to an entry in the frequency table. The contents of the entry are applied to the selection lines of the MUX and dynamically select the appropriate clock frequency for the resource.

Thus, the clock frequency for the resource is automatically determined by the total bandwidth utilization of the controllers requesting access to the resource. Accordingly, the clock frequency is preferably chosen so that the bandwidth of the resource closely matches the needed bandwidth. As a result, little power is wasted due to operating the resource at a higher clock frequency than is necessary. A preferred embodiment of the present invention further conserves power by shutting off the memory clock when no controllers are requesting access to the memory controller.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a high-level block diagram illustrating the components of a portable electronic device according to an embodiment of the present invention;

FIGURE 2 is a block diagram illustrating internal functional units of an ASIC in a portable electronic device according to one embodiment of the present invention;

FIGURE 3 is a block diagram illustrating the various clocks used by an embodiment of the present invention;

FIGURE 4 is a block diagram illustrating circuitry within the system clock controller of FIG. 3 for determining a clock frequency;

FIGURE 5A is a block diagram illustrating circuitry within the system clock controller for deriving and selecting clocks from a master clock;

FIG. 5B is a circuit diagram illustrating a multiplexer for selecting an output clock from among multiple input clocks;

FIGURE 6 is a flow chart illustrating the operation of the ASIC according to an embodiment of the present invention when selecting an appropriate clock frequency for a resource; and

FIGURE 7 is a block diagram illustrating a lower-level view of the interface between the memory controller and the external memory according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is a high-level block diagram illustrating the components of a typical portable electronic device 100 according to an embodiment of the present invention. As

used herein, the phrase "portable electronic device" refers to the class of devices including smart telephones, screen telephones, and other Internet-ready appliances. The phrase also includes devices such as digital cameras, portable music players, and hand-held computer systems. Of course, the present invention has broad applicability and can be used with any electronic device where power conservation is desired, regardless of whether the device is portable.

FIG. 1 illustrates an application specific integrated circuit (ASIC) 110, preferably manufactured using a complementary metal oxide semiconductor (CMOS) process, and having a central processing unit (CPU) 112, a digital signal processor (DSP) 114, a direct memory access (DMA) controller 116, and a memory cache 118. Although a preferred embodiment of the present invention utilizes a CMOS manufacturing process, other embodiments can use other processes, including: bipolar CMOS, transistor-transistor logic, and emitter coupled logic. In one embodiment, the CPU 112 is an ARM720TDSP microprocessor and performs the processing for the portable electronic device 100. This processor is a 32-bit reduced instruction set computing (RISC) processor having 16-bit instruction extensions supporting low-cost consumer devices.

The DSP 114 provides a digital signal processing engine for applications such as speech recognition, data compression, modem and other forms of communication, and other real time applications where complex signal processing is desired. In one embodiment, the cache 118 is an 8 Kilobyte (KB) unified cache with a 64-entry translation lookaside buffer (TLB). Preferably, the DSP 114 and cache 118 are integrated into the CPU 112 core in order to enhance the performance of the CPU.

The portable electronic device 100 also preferably includes external memory devices 120 acting as the main memory for the portable electronic device 100. The amount of memory can vary depending on the functions performed by the device 100. In a preferred embodiment of the present invention, the external memory devices 120 are synchronous dynamic random access memories (SDRAMs) compliant with the PC100 specification.

The DMA controller 116 preferably allows peripherals to access the external memory 120 without requiring CPU 112 intervention. Thus, the DMA controller 116 allows fast data transfers while the CPU 112 is busy with other tasks or even in a low-power standby mode.

Peripherals that can be associated with the portable electronic device 100 include a display device such as a monochrome liquid crystal display (LCD) panel 122 and/or a color LCD panel 124 for displaying information to a user and an input device such as a keyboard 126, mouse, and/or touch screen for receiving information from a user or another device. The portable electronic device 100 may also have an associated codec 128 for performing digital to analog and analog to digital signal conversion. Typically, the codec 128 converts an analog signal from a microphone 130 into a digital signal and converts digital signals stored in the device 100 into analog signals for output to a speaker 132. The codec 128 may also be coupled to a digital audio adapter (DAA) 134 for transmitting and receiving digital audio data.

In addition, the portable electronic device 100 may have one or more interfaces for accepting function cards 136. These cards include smart cards, MMC cards, Personal Computer Memory Card International Association (PCMCIA) cards, and flash memory

cards providing functionality such as additional memory storage, audio codecs, software, display interfaces, electronic commerce transactions, and any other functionality that can be included on a card.

The portable electronic device 100 also preferably includes a power source 138. The source 138 may be, for example, common consumer-sized batteries such as AA or AAA batteries, or more specialized batteries using lithium-ion or nickel-metal-hydride (NiMH) technology. In addition, the power source may be a constant power source such as a transformer coupled to a wall outlet. In one embodiment, the present invention draws approximately 50-200 microamps of current during normal use.

FIG. 2 is a block diagram illustrating the internal functional units of the ASIC 110 according to one embodiment of the present invention. The CPU 112 contains the cache 118 and is coupled to the DSP 114 and a system bus 210. The system bus 210 couples a memory controller 212, an on-chip static random access memory (SRAM) 214, an interrupt controller 216, a peripheral bridge 218 and a PCMCIA interface 220.

The memory controller 212 controls accesses to and from the memory 120 by the CPU 112 and the assorted controllers and peripherals. The memory controller 212 also controls power consumption by the memory 120 and places the memory 120 in a low-power mode when there are no memory requests, as described below.

The memory controller 212 is preferably coupled via a memory bus 213 to a color/monochrome LCD controller 222, a monochrome LCD controller 224, a multi ICE interface 226, and the DMA controller 116. The color/monochrome LCD controller 222 preferably supports large displays such as thin film transistor (TFT) and dual scan supertwist nematic (DSTN) displays while the monochrome LCD controller 224

preferably supports smaller, lower resolution displays. Alternative embodiments of the present invention may contain controllers for other display technologies, such as cathode ray tubes (CRTs) and gas plasma.

The multi ICE interface 226 allows the on-chip CPU to be controlled by an external device for debugging purposes. The DMA controller 116 provides direct access to the memory as described above with respect to FIG. 1.

The SRAM 214 preferably contains a relatively small amount of memory that remains valid even when the external memory 120 is powered down. In one embodiment, the SRAM 214 holds approximately 5 KB of memory and acts as a frame buffer for the monochrome LCD display 122. The interrupt controller 216 processes interrupts generated by the other devices coupled to the system bus 210.

The peripheral bridge 218 is coupled via a peripheral bus 228 to one or more universal asynchronous receiver/transmitters (UARTs) 230, one or more MMC card interfaces 232, one or more synchronous serial interfaces 234, a general purpose I/O interface 236, and the DMA controller 116. The UARTs 230 preferably interface with external devices receiving serial data, such as wireless communication devices and modems. The one or more MMC interfaces 232 preferably interface with MMC cards for providing solid state storage to the portable electronic device 100. The one or more synchronous serial interfaces preferably interface with devices such as modems, touch screen controllers, and codecs. Finally, the general purpose I/O allows the CPU 112 to drive or sample external logic. The DMA controller 116 provides the devices on the peripheral bus 228 with direct access to the memory controller 212 without direct CPU 112 control.

The system bus 210 is also coupled to a static memory interface 220. The static memory interface 220 can interface with devices like read only memory (ROM), SRAM, flash memory, and PCMCIA 2.0 compliant devices.

The ASIC 110 preferably contains first and second oscillators 238, 240 and a timer/real time clock (RTC) 242. The oscillators 238, 240 are used by one or more programmable phase locked loops (PLLs) to generate the reference clocks for the system clock controller, as described in more detail below. The timer/RTC 242 preferably provides two 16-bit timers and a 32-bit RTC for use by the ASIC 110.

The ASIC 110 also preferably contains a power management controller (PMC) 244. The PMC 244 can preferably place the portable electronic device 100 in one of five power states: run mode, when all functions are enabled; idle mode, when the CPU clock is shut off; snooze mode, when the LCD 122 is refreshed from the on-chip 110 SRAM 214; standby, when the portable electronic device 100 is in a low power mode but can make a quick transition to run mode; and deep sleep, an ultra low power mode. The latter four modes conserve power in the ASIC 110 when not in run mode.

FIG. 3 is a block diagram 300 illustrating the various clocks used by an embodiment of the portable electronic device 100 according to the present invention. At least one programmable PLL 310 uses one of the oscillators 238, 240 shown in FIG. 2 and produces a master, or reference, clock signal at a selected frequency. The master clock is transmitted to a system clock controller 312 which uses the master clock to generate the different clocks used to run the portable electronic device 100. In one embodiment of the present invention, there is one PLL generating a master clock for the whole ASIC 110.

In one embodiment of the present invention, the system clock controller 312 generates a memory clock 314 having a frequency of approximately 100 MHz. The memory clock 314 is passed to and synchronizes data accesses between the memory controller 212 and the external memory devices 120.

The system clock controller 312 also generates a bus clock 316 having a frequency of approximately 50 or 100 MHz. The bus clock is passed to the three busses 210, 213, 228, collectively illustrated as bus 322 in FIG. 3, and synchronizes data transfers among the various devices coupled to the bus 322.

The system clock controller 312 also preferably generates one or more peripheral clock 318 having frequencies ranging from approximately 32 kHz to 60 MHz. The peripheral clocks 318 are used by the peripheral controllers to interface with the external peripheral devices.

In addition, the system clock controller 312 preferably generates a CPU 112 clock having a frequency in the range of 3.68 to 100 MHz that is passed to the CPU 112. Finally, the system clock controller 312 preferably generates a clock controlling the operation of the clock controller 312 itself. All of the frequencies above describe one embodiment of the present invention and alternate embodiments may utilize different frequencies. Moreover, these frequencies can dynamically vary depending upon the needs of the ASIC 110 at a given instant in time as described in detail below.

The various devices in the ASIC that can be accessed by other devices in the ASIC 110, such as the bus 322, the memory controller 212, and the on chip SRAM 214, are referred to as "resources." The speed at which a resource is clocked has a direct effect on the rate at which the resource can process data (i.e., the bandwidth of the resource).

For example, the memory controller can process data at a higher bandwidth when it is clocked at a higher frequency than it can when it is clocked at a lower frequency.

The devices in the ASIC 110 that can access a resource are referred to as "controllers." For example, a UART controller 230 is a controller because it can access the memory controller 212 via the DMA controller 116. Typically, a controller will access a resource within a given bandwidth range. For example, the UART may send or retrieve data from the memory controller at a low bandwidth relative to the bandwidth supported by the memory controller when it is clocked at its maximum rate. When multiple controllers concurrently access a resource, the resource should support a bandwidth equal to the sum of the bandwidth utilization of the controllers.

Each controller in the ASIC 110, including the CPU 112 and the peripheral controllers, of which controllers 222 and 230 are representative, preferably has request lines 324, 326, 328 coupled to the system clock controller 312. In one embodiment of the present invention, each controller has separate request lines to the system clock controller 312 for each resource in the ASIC 110, such as the bus 322 or the memory controller 212, that the controller can access. In another embodiment, each controller has a single request line to the system clock controller 312 for requesting access to any resource. Other embodiments may have different numbers of request lines, or even a different scheme for a controller to signal the system clock controller 312 that the controller will utilize a resource. Other embodiments may lump multiple requests together and generate a single request to indicate access to a resource.

A controller desiring access to a resource such as the memory controller 212 or the bus 322 preferably enables the request line before accessing the resource. In addition,

the controller preferably leaves the request line enabled until the controller no longer requires accesses to the resource. As described below, the system clock controller 312 automatically and dynamically determines an appropriate clock rate for the resource from the enabled request lines.

FIG. 4 is a block diagram illustrating circuitry 400 for dynamically determining a clock frequency for a resource within the ASIC 110. As used herein, the term "dynamically" means that the clock frequencies for resources are determined and varied while the clocks and resources are running. The frequency variation happens automatically, meaning it happens without interrupting the CPU 112. Accordingly, the CPU 112 does not waste time determining whether to vary a clock frequency and the clock frequency is varied proactively. In this manner, the present invention saves power by lowering clock frequencies yet does not sacrifice performance.

Preferably, the circuitry 400 for dynamically choosing a clock frequency resides within the system clock controller 312 of FIG. 3. Illustrated are three request lines 410A, 410B, 410N labeled "1" through "n" to represent the n request lines received from the n ASIC 110 controllers such as the CPU 321 and peripheral controllers 222, 230. The request lines 410 illustrated in FIG. 4 generally correspond to request lines 324, 326, and 328 shown in FIG. 3.

In addition, three bandwidth registers 412A, 412B, 410N labeled "1" through "n" correspond to the n request lines 410. Thus, in one embodiment of the present invention every request line 410 to the clock controller 312 also has an associated bandwidth register 412. In other embodiments of the present invention, multiple request lines 410 may be associated with a single bandwidth register 412.

The bandwidth registers 412 are preferably software programmable and hold values indicating the bandwidth utilized by the controller or controllers driving the associated request line or lines 410. In one embodiment, the registers 412 hold 8-bit values. A relatively low value in a register 412 indicates that the associated controller utilizes relatively low bandwidth while a relatively high value indicates that the associated controller utilizes relatively high bandwidth. For example, the CPU 112 is likely to use more bandwidth than a UART 230 so the bandwidth register associated with the CPU 112 will have a higher value than the bandwidth register associated with the UART 230. Since the registers 412 are software programmable, the values held in the registers 412 and the interpretations given to those values can vary depending upon the needs of the software. In other embodiments, the bandwidth register 412 can be a hard-wired value if flexibility is not a concern.

When a request line 410 is enabled, the value of the associated bandwidth register 412 is passed through a decoder 414 to an adder input register 416 for synchronizing input to an adder 418. When a request line 410 is not enabled, the decoder 414 passes a value of zero to the adder input register 416. When triggered by the clock, the adder input registers 416 pass their contents to the adder 418. The adder 418 sums the values in the adder input registers 416 and produces an output sum. Thus, the output of the adder 418 is the sum of the contents of the bandwidth registers 412 having enabled request lines 410. Although a preferred embodiment of the present invention uses an adder 418 to sum the contents of the bandwidth registers 412, any circuitry for adding values can be substituted. For example, the functions of the adder 418 can be performed by an arithmetic logic unit (ALU) in the system clock controller 312 or dedicated logic.

The output of the adder 418 is passed to a holding register 419. The registers before 416 and after 419 the adder 418 allow the adder 418 to utilize one full clock cycle to sum the inputs. If the adder 418 requires additional clock cycles to calculate the sum, additional stages of registers can be added after the adder 418. The last stage register that holds the sum of the bandwidth registers 412 is the holding register 419.

Then, the sum in the holding register 419 is used as an index to an m-entry frequency table, where each entry corresponds to a clock frequency derived from a particular master clock by the system clock controller 312. The contents of the selected entry in the frequency table 420 are passed through additional stages of registers to allow for the access time needed to retrieve the value from the frequency table 420. In the illustrated embodiment, there are two stages of registers 421, 422. The last stage register that holds the output of the frequency table 420 is the clock MUX select register 422. The contents of the clock MUX select register 422 are used to select a particular clock frequency for a resource as described below. Preferably, the selected clock frequency is the lowest frequency necessary to support the bandwidth utilized by the controllers requesting access to the resource.

In one embodiment of the present invention, the system clock controller 312 contains multiple instances of the circuitry illustrated in FIG. 4, with each instance associated with a particular resource or clock on the ASIC 110. For example, one instance may be used to select the frequency of the memory clock 314, another instance may be used to select the frequency of the bus clock 316, and another instance may be used to select the frequency of the CPU clock 320. While peripheral clocks 318 can also be selected in this manner, in practice many of the peripherals use fixed clocks.

FIG. 5A is a block diagram illustrating circuitry 500 within the system clock controller 312 for deriving clocks from a master clock and selecting a clock. As described with respect to FIG. 3, the PLL 310 generates a master clock. The circuitry 500 contains at least m clock dividing modules 510, and each module derives a lower frequency clock from the master clock. For example, in the embodiment illustrated in FIG. 5, the circuitry 500 generates clocks having frequencies of ½, 1/3, ¼, down to 1/m of the master clock. In addition, the circuitry 500 preferably contains one or more clock dividing modules 512 for deriving specific clock frequencies for those peripherals that require fixed frequency clocks. The circuitry 500 outputs the master clock as well as the derived lower-frequency clocks.

FIG. 5B is a circuit diagram illustrating a multiplexer (MUX) 514 for selecting an output clock from among the multiple clocks generated by the circuitry 500 of FIG. 5A. The multiple clocks are passed to the MUX 514. In addition, the output of the clock MUX select register 422 holding the entry from the frequency table 420 is applied to the select lines of the MUX 514. As a result, the MUX 514 selects and outputs the clock having the frequency determined by the selected entry in the frequency table 420 of FIG. 4. Although a preferred embodiment of the present invention utilizes a MUX 514 to select the appropriate clock frequency, any circuitry for selecting a clock in response to the entry in the frequency table 420 can be utilized instead. In addition, in an alternative embodiment the output of the adder 418 is applied directly to the MUX 514 and the MUX 514 is configured to select the proper clock in response.

With the exception of the clock MUX select register 422, the various registers 412, 414, 416, 419, 421 illustrated in FIG. 4 are preferably clocked on the rising edge of

the clock signal. The clock MUX select register 422 is preferably clocked on the falling edge of the clock signal because the various lower-frequency clocks are derived from the rising edge of the master clock. When a clock having a different frequency is selected by the clock MUX 514, there might be a small delay while the selection logic switches clocks. If the clock MUX 514 is switched right after the rising edge of the clock, then this delay may produce an undesirable glitch on the output of the clock MUX 514. If the clock MUX 514 is switched on the falling edge of the clock, then this allows half a clock period for the delay between the individual clocks. Conversely, if the lower-frequency clocks are based on the falling edges of the clock, then the clock MUX select register 422 should be based on the rising edge of the clock.

In a preferred embodiment, the system clock controller 312 has at least one instance of the circuitry 500 of FIG. 5A for each master clock generated by the ASIC 110. Likewise, the system clock controller 312 preferably has at least one instance of the circuitry illustrated in FIG. 5B for each frequency table 420. In this manner, the system clock controller 312 can simultaneously select different clock frequencies for any resource in the ASIC 110, including the memory 314, bus 316, peripheral 318, and CPU 320 clocks.

FIG. 6 is a flow chart illustrating the operation of the ASIC 110 according to a preferred embodiment of the present invention when selecting an appropriate clock frequency for a resource in the ASIC 110. The steps illustrated in FIG. 6 may be independently performed for each clocked resource in the ASIC 110. For example, the steps can be used to independently select the clocks for the memory controller 212 and the system bus 322.

If any controller in the ASIC 110 needs access 610 to a resource such as the system bus 322 or the memory controller 212, the controller signals 612 the system clock controller 312 by enabling the appropriate request line or lines 410. The system clock controller 312 sums 614 the contents of the bandwidth registers 412 corresponding to the request lines 410 activated by the controller or controllers. The resulting sum is used to look up 616 an entry in the frequency table 420 to select 618 a clock frequency for the requested resource. The selected frequency 618 is preferably the lowest possible clock frequency for the resource that is adequate to meet the bandwidth requirements of the controllers requesting access. Of course, the values held in the bandwidth registers 412 or the frequency table 420 can be adjusted to meet other requirements of the ASIC 110. For example, it may be desirable to select a clock frequency that provides slightly more or slightly less bandwidth than would be necessary under ideal conditions in order to account for real world performance considerations. The contents of the selected entry in the frequency table 420 are preferably applied to a MUX 514 in the system clock controller 312 and select the corresponding clock frequency derived from the master clock.

If no controller in the ASIC 110 is requesting access 610 to a resource, then the sum of the bandwidth registers calculated by the system clock controller 312 is zero. Preferably, the corresponding entry in the frequency table 420 lowers 620 the clock to its minimum operating frequency or stops the clock altogether.

If there are no requests to access the memory controller 212, then the system preferably stops 622 the memory clock from toggling. FIG. 7 is a block diagram illustrating a lower-level view of the interface between the memory controller 212 and the

external memory 120. FIG. 7 illustrates five communications lines, labeled "CLK" 710, "CKE" 712, "CNTRL" 714, "ADDR" 716, and "DATA" 718. As well understood in the art, the CNTRL 714, ADDR 716, and DATA 718 lines respectively pass control signals between the memory 120 and the memory controller 212, send the address information to the memory 120, and pass data between the memory 120 and the memory controller 212. The CLK 710 line passes the memory clock 314 to the memory 120 while the CKE 712 line typically tells the memory 120 when the clock on the CLK 710 line is enabled. When the CKE 712 line is disabled, the memory 120 places itself in a low-power mode. In a preferred embodiment of the present invention, when there are no requests to access external memory 120, the memory controller 212 disables the CKE 712 line and the system clock controller 312 stops the memory clock. Accordingly, no power is wasted on the CLK 710 line when the memory 120 is disabled by the CKE 712 line.

Accordingly, the present invention minimizes power usage by the portable electronic device 100 by reducing the clocks of the ASIC 110 to the minimum frequency required to support the bandwidth utilized by the controllers requesting access to a resource. Indeed, one embodiment of the present invention reduces power use by almost 70% compared to a system having fixed clock speeds. The present invention achieves these goals without sacrificing performance or requiring major alterations to the system design.